POINT MATCHING METHOD FOR FLAW DETECTION IN PRINTED CIRCUIT BOARDS

M. J. Donahue, A. P. Sprague and S. I. Rokhlin

Department of Welding Engineering
The Ohio State University
190 W. 19th Avenue
Columbus, OH 43210

INTRODUCTION

There is high demand in industry for automated inspection of printed circuit boards (PCB's). The major requirements for inspection systems are high speed, universality and high precision of inspection. Two major approaches have been used in the past: the reference-comparison and the design-rule verification methods. Each of them has its limitations and advantages. Recently, several authors have described algorithms which integrate both the reference-comparison and the design-rule methods [1-4].

In this paper we describe an algorithm which includes elements of the reference-comparison and the design-rule methods. It is distinguished by relative simplicity of implementation and high efficiency. In our method, a set of the directed (segment) graphs is first created from the image of the PCB; then the feature points are extracted from the segment graphs via a rule-based algorithm. The set of feature points includes wire intersections, pads, circuit defects, etc. Finally, the set of feature points is matched to the reference set. The implementation of the algorithm is demonstrated here by several examples.

EXPERIMENTAL SYSTEM

The schematic diagram of the experimental system is shown in Fig. 1. The images are picked up with a TV camera. Next they are digitized using a Data Translation IBM compatible arithmetic frame grabber. This board is also used for data transfer and data representation on the monitor. A Definicon parallel processing board with four 20 MHz INMOS T800 transputers and 1 megabyte of RAM for each transputer are used for high-speed data processing. The transputers are programmed in the C language in the parallel processing mode. The number of transputers and therefore the parallel processing power may be increased as required. The digitizing and processing boards are housed in an IBM-AT compatible.

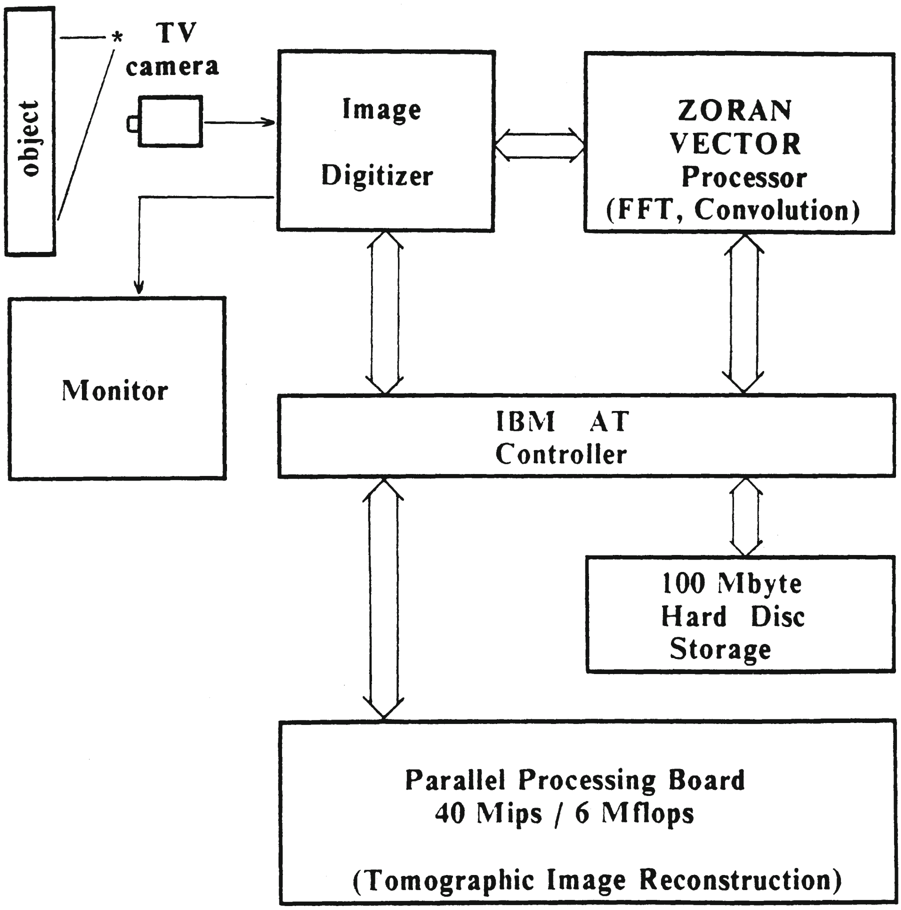

Fig. 1 Schematic diagram of the experimental system

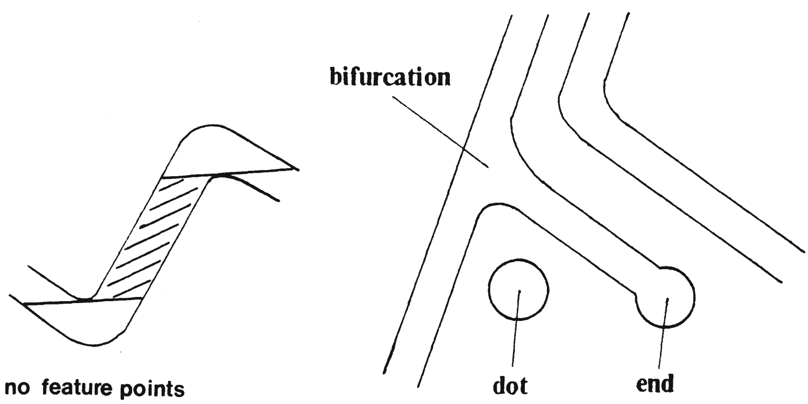

Fig. 2 Definition of feature points in a circuit

DESCRIPTION OF THE METHOD AND RESULTS

Method Outline

The algorithm consists of two major parts: encoding and matching. The encoding algorithm scans digitized images of the PCB and extracts specific elements which are called 'feature points'. The matching algorithm compares the file of points obtained by the encoding algorithm with the reference file and identifies additional or missing feature points which are associated with circuit defects.

The scanning algorithm performs a row-by-row scan of the image, and produces a list of feature points. We define as feature points such elements of the circuit as pads, wire intersections, metal dots, and wire ends. Circuit defects such as wire breaks, bridges (shorts) between wires, and pores inside wires are also considered as feature elements. Examples of feature points in the circuit are shown schematically in Fig. 2. Defects of printed circuits are also represented by the feature points during encoding as illustrated in Fig. 3.

The scanning algorithm does not extract feature points directly; instead, an intermediate, much more compact representation is first generated. The intermediate representation is a directed graph, called the segment graph.

Intermediate Data Representation: The Segment Graph

The scanning algorithm creates a structure, which we call the segment graph, in a three-step process. The first step is the extraction of objects called 1-runs from individual rows. The second step is a combining of these 1-runs into sets called segments. The last step is an ordering of these segments, which is the segment graph.

Since (after thresholding) the value of the digitized image at each pixel is zero or one, the sequence of values of pixels in a row is a string of zeroes and ones. We say that a 1-run in a row is a sequence of adjacent pixels, all with value 1. A maximal 1-run is a 1-run which is not contained in a longer 1-run. For example, in the sequence 011110011111, the third through fifth pixels form a 1-run which is not maximal, since the second through fifth pixels form a longer 1-run containing the former. This string contains two maximal 1-runs, one of length 4 and one of length 5. We define 0-runs in a like manner. Since the metal is lighter than the background in most images, we are principally interested in 1-runs. Moreover, we will only be interested in maximal runs, so henceforth the term '1-run' is implied to mean 'maximal 1-run'. We use the notations $R(j)$ to denote a 1-run in row j.

Our algorithm compares 1-runs of adjacent rows. A 1-run of row j matches a 1-run of row j-1 if:

1. The runs overlap, i.e. have one or more columns in common.
2. Neither overlaps any other 1-runs of these two rows.

A segment is a collection of 1-runs $R(s)$, $R(s+1)$, ..., $R(t)$ such that:

1. $R(j)$ is a 1-run in row j ($s<=j<=t$)
2. $R(j)$ and $R(j-1)$ are matched ($s+1<=j<=t$)
3. The collection is maximal with respect to 1 and 2.

Figure 4(a) is an example of a metallization pattern with 19 segments as indicated.

The segment graph of a PCB is a directed graph on the set of segments. A directed graph is a pair (V,A) where each element of V is called a vertex

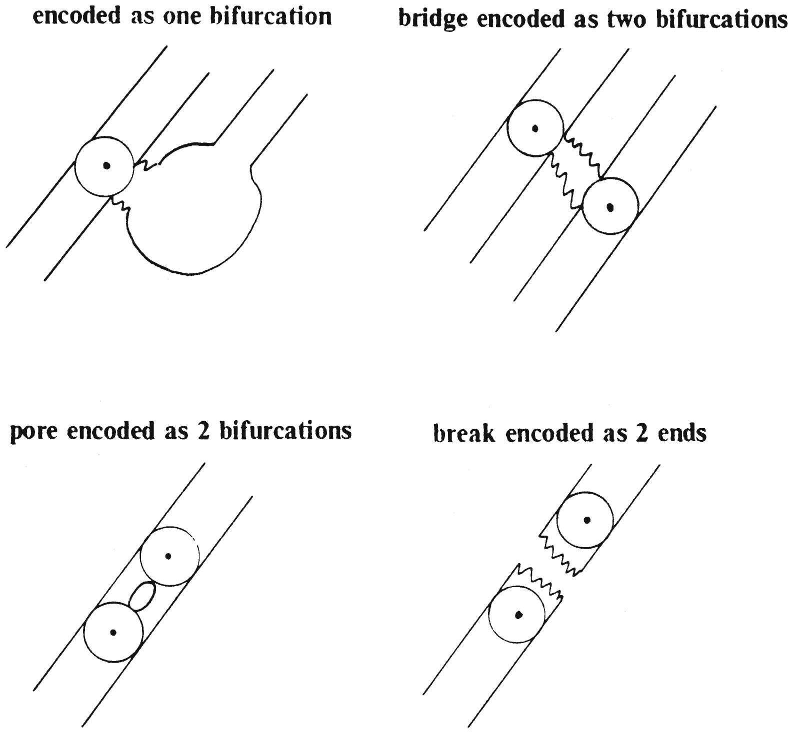

Fig. 3 Representation of defects by feature points.

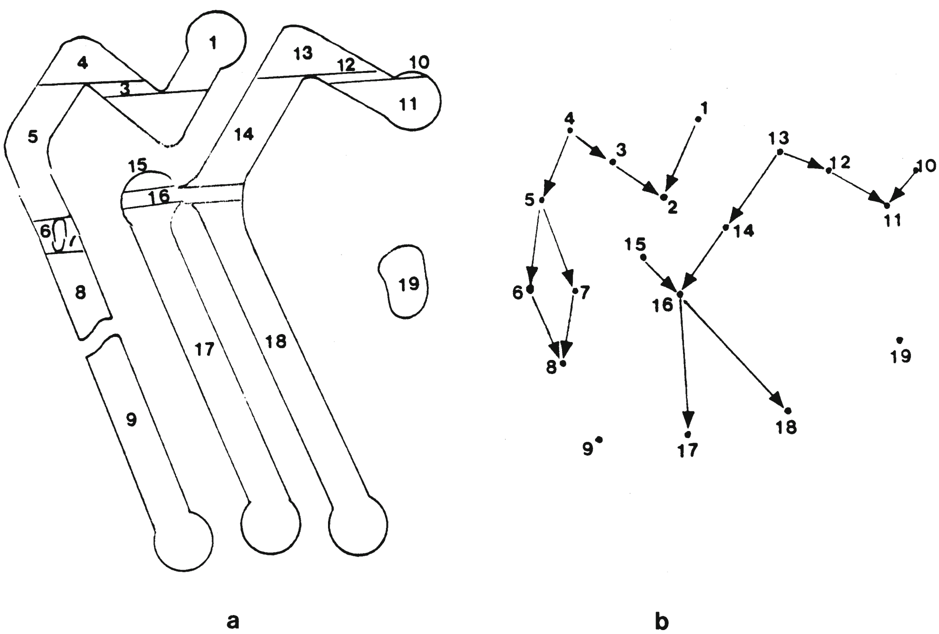

Fig. 4(a) Fragment of electric circuit and
(b) its segment graph representation.

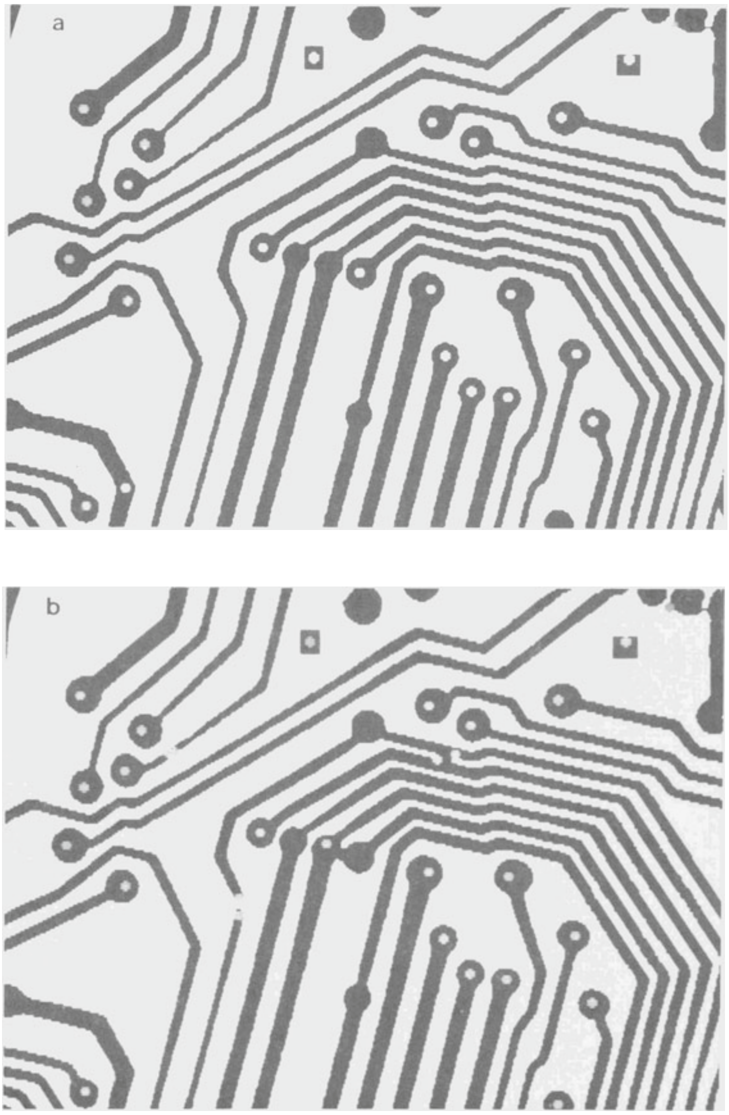

Fig. 5(a) Section of a reference board with the extracted feature points indicated by gray points.

(b) Section of a tested board with defect. Extracted feature points also indicated by gray points.

or a node, and each element of A is called an arc. Each arc is an ordered pair of elements of V. A diagram depicting a directed graph typically represents vertices as dots and arcs as arrows between dots.

To form the segment graph, let the segments of the PCB be vertices, and draw an arc from segment S= R(p), R(p+1), ..., R(q) to segment S'= R'(s), R'(s+1), ..., R'(t) if the lowest 1-run in S overlaps the highest 1-run in S', i.e. if R(q) overlaps R'(s). Note that R(q) does not match R'(s) (by maximallity of the segments), so it must be that either R(q) overlaps more than one 1-run of row s or else R'(s) overlaps more than one 1-run of row q. The segment graph of Fig. 4(a) is displayed in Fig. 4(b).

Extraction of Feature Points

Feature points are extracted from the segment graph via a rule based algorithm. Each end (upper and lower) of a segment is a potential feature point location. The decision to place a feature point at an end is governed by the size of the segment and the number of neighboring segments at each end (neighboring segments are those with an arc between them). For example, suppose a segment S has two neighbors at the top and no neighbors at the bottom. If S is large enough, then two features points are extracted; a bifurcation at the top and a end feature at the bottom. On the other hand, suppose S has only one neighbor at the top and one neighbor at the bottom. Then S is a continuation of a metal strip and so no feature points are extracted.

The extraction algorithm examines each segment of the PCB. Fig. 5(a) is a section of a reference board with the extracted features indicated by solid circles. Fig. 5(b) is the same section of a test board with its features similarly marked.

Fig. 6 Feature points extracted from the reference board

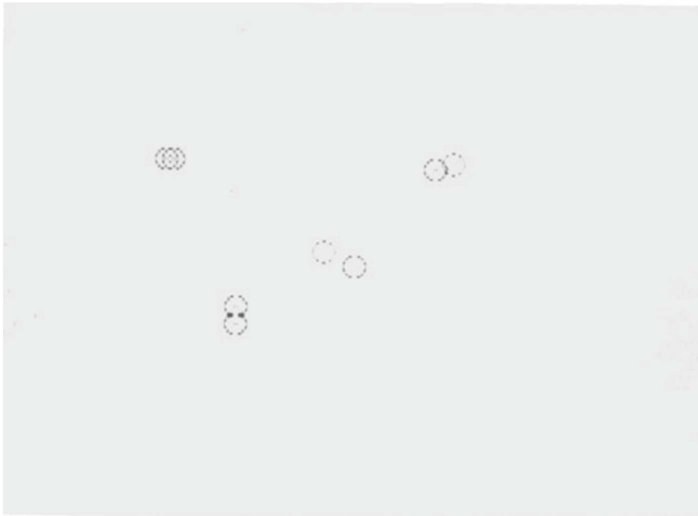

Fig. 7 Display of unmatched feature points which correspond to the defects.

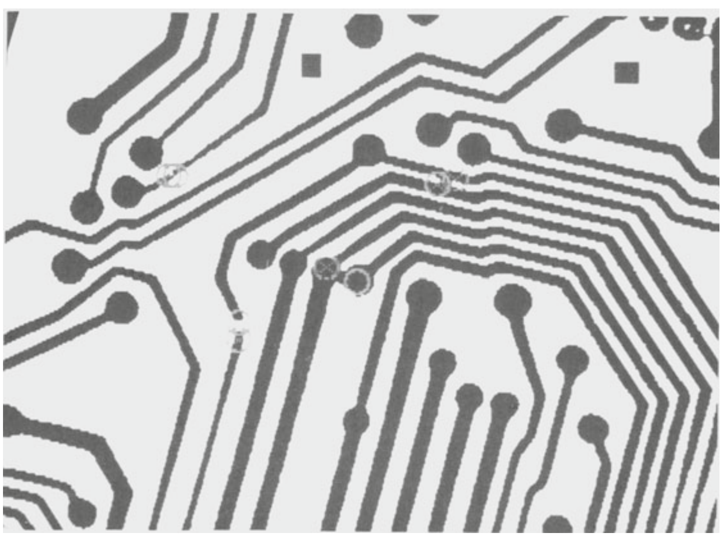

Fig. 8 Overlay of unmatched feature points from Fig. 7 on an image of the test board.

Comparison Algorithm

The comparison algorithm works on the extracted feature point data. Fig. 6 shows the feature points extracted from the reference board of Fig. 5 (a). A similar data set is available for the test board. The goal of the comparison algorithm is to put these two sets of feature points into a one-to-one correspondence. It is important to note that the comparison involves only the extracted feature point data, which is a very small set compared to the original images which have millions of pixels each.

The first step of the comparison algorithm is an alignment phase. Special points introduced on the actual PCB's are used to 'line up' the two data sets, so that the test board data may be considered to be overlaid on top of the reference board data.

The second step is to match features from the test board to features form the reference board. A feature X from the test board is said to match a feature Y from the reference board if they are of the same type (i.e. bifurcation, feature ending, etc.) and if the distance between X and Y is less than a fitting tolerance epsilon. This fitting tolerance allows small distortions in the test board, but must be chosen smaller than half the distance between any two features of the reference board to insure that no feature point from the test board can match more than one feature on the reference board. This constraint guarantees uniqueness of the match.

Our algorithm also divides the reference board into small rectangular search blocks to enhance matching speed. We can match two sets of 5000 features each in under 1 second.

The output of the comparison algorithm is a list of those features on each board which were left unmatched. Fig. 7 displays this output. In this figure, the circled crosses represent unmatched features from the test board, and the empty circle is an unmatched feature from the reference board. Fig. 8 shows the previous figure overlaid on an image of the test board.

ACKNOWLEDGEMENT

This work was sponsored by the Center for Advanced Nondestructive Evaluation, operated by the Ames Laboratory, USDOE, for the Air Force Wright Aeronautical Laboratories/Materials Laboratory under Contract No. W-7405-ENG-82 with Iowa State University.

REFERENCES

1. L. F. Pan, IEEE Trans. Pattern Anal. and Mach. Intel. PAMI-5, 602-608 (1983).
2. Y. Hara, N. Akiyama, and K. Kapasaki, IEEE Trans. Pattern Anal. and Mach. Intel. PAMI-5 623-630 (1983).
3. G. A. W. West, IEEE Trans. Syst., Man, Cyber, SMC-14, 767-773 (1984).
4. J. R. Mandeville, IBM J. Res. Develop. 29, 73-86 (1985).